Microsoft Naming Convention for "Hidden" Names

Allen Akin

Microsoft Corporation

1. BACKGROUND

Our compilers generate references to hundreds of "hidden" names – runtime routines, special external variables, and the like. Neither our applications nor our users' programs may define a publicly-visible name that conflicts with one of these hidden names, because that would break the associated language function.

Furthermore, in mixed-language programs, no hidden name used by one language may conflict with a hidden name used by another. With the advent of CMERGE and Windows, mixed-language programs are becoming quite common; for example, any FORTRAN program now contains runtime code written in C and Pascal as well as the main program in FORTRAN.

Looking to the future, we might want to guarantee that there are no conflicts with hidden names defined for particular environments, like XENIX, Windows, or DOS. As our products become more highly integrated, we may also want to prevent conflicts across applications.

Up to this point, each product development group has insured that there are no name conflicts in its products. There has been little need for communication between groups on this issue. Now that products are being merged, the possibility of naming conflicts is significantly greater, and it's unlikely our testing regimen will uncover all such conflicts in all possible combinations of products.

By adopting a naming convention now we can reduce the chance that naming conflicts will arise in the future.

This document contains:

- A naming convention for hidden names.
- Instructions for using the convention to devise new names.
- A list of exceptional names that must be avoided.
- The history of this particular convention.
- A short list of pitfalls for the unwary.

2. THE NAMING CONVENTION

1. This convention applies only to "hidden" names.

   Documented user-visible names, such as many conventional C library routines, need not follow it. There is always the possibility that user-visible names may clash; this possibility can be minimized by following this convention for new user-visible names. (I have some reservations about recommending such a policy.) No public name should have the form of a hidden name unless it meets all the requirements of this convention.

2. Everyone should follow this convention.

   If a product is released with non-conventional names, later we may not be able to integrate it with another product and retain upward compatibility. Following this convention should make integration easier and less risky.

3. A hidden name for a given language or environment begins with a prefix that is unique to the language or environment.

   The following table defines the prefixes:

Language/Environment	Prefix
Common to All	__a
BASIC	__b
C	__c
COBOL	__k
DOS	__d
FORTRAN	__f
Pascal	__p
Windows	__w
XENIX	__x

   Note that *if you are programming in C, the names will begin with just one underscore.* Also note that *letter case is significant.* For comments concerning the "Common to All" case, see "PITFALLS" below.

4. Within a given language or environment, the rules for devising names are up to the group responsible for the language or environment.

   A group may elect to follow additional conventions for the portion of a hidden name after the prefix. For example, the name could be based on Hungarian, or on the memory model, or on the version number of the associated product. It may also be reasonable not to follow a convention at all. Such matters are beyond the scope of this document.

5. A hidden name must not appear in the list of exceptions.

   The exceptions are listed in section 4. Each exception is a name that already exists somewhere in the runtime of a language or environment. By avoiding these names we protect upward compatibility and reduce our future software maintenance effort.

6. If necessary, a group must establish its own policy to insure that there are no name conflicts within a given product.

3. USING THE NAMING CONVENTION

When you need to devise a new name, do the following:

1. Select a name, based on the prefix appropriate to your product and any special conventions internal to your group.
2. Verify that the name does not appear in the list of exceptions. If it does, you must choose an alternative name.
3. Take whatever actions are necessary to insure that no one else in your group has also chosen that name.

4. EXCEPTIONS

The following names are already in use in by at least one language or environment. They must be avoided when selecting new names, either hidden or public. Please note that *in C, these names would be written with a single leading underscore,* and *letter case is significant.*

```
__3to4
__ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
__AMP          __AST          __AT           __AUTO         __Acttab
__BAR          __BASENAME_    __BLNK_        __BLNK__       __BRA
__BREAK        __BSL          __BSS          __BUFSYNC      __Bit
__C            __CBR          __CKT          __CSTART       __DATA
__DEFINE       __DIG          __DOL1         __DOL2         __DORIGIN
__DQU          __EDATA        __EF           __END          __EOF
__EOR          __EQ           __ETEXT        __FBSS         __FDATA
__FILENAME     __FILE__       __FOR          __GT           __HAT
__IDCH         __ID_          __IF           __IOEOF        __IOREAD
__IORW         __IOWRT        __KET          __L            __LINE__
__LPC          __LQU          __LT           __MAIN         __MAIN__
__META         __MIN          __MIN_BLKS     __N            __NFILE
__NUM          __O            __OPTIONS      __Objent       __Objtab
__Objtmp       __Objval       __P            __PCS          __PLS
__POP_         __QU           __RETURN       __S            __SEM
__SH_          __SORIGIN      __SPC          __START        __Sigtramp
__Spacct       __Syspc        __TAB          __TEXT         __TORIGIN
__TTY          __TV           __TVORIG       __U            __UNIvec
__UPC          __V            __WHILE        __X            __a
__acsjmptab    __active       __addr         __adm          __advance
__aintr        __aldiv        __alloca       __almul        __alrem
__alshl        __alshr        __aop__        __arg          __asnjmptab
__assert       __at           __atn2jmptab   __atnjmptab    __auldiv
__aulmul       __aulrem       __aulshr       __b            __base
__bdevcnt      __bdevsw       __bits         __blanks       __bldiv
__blkptr       __blmul        __blocks       __blrem        __blshl
__blshr        __bool_        __braelist     __branch       __braslist
__bravar       __brkctl       __buf          __bufend       __bufendtab
__bufsyn       __bufsync      __buldiv       __bulmul       __bulrem
__bulshr       __byte         __bytptr       __c_clea       __c_why_
__callout      __catch_       __cclass       __cdir         __cflg
__cheat        __chk_ty       __chkln        __chkstk       __chkstk4
__cksum        __cleanu       __cleanup      __clearh       __clearl
__clock        __clreol       __cnt          __combop       __compha
__comptr       __con_tty      __cond         __const        __coremap
__coshjmptab   __cosjmptab    __cost_f       __countb       __countbase
__countc       __countend     __cron         __cropzeros    __ct
__ctrandisp1   __ctrandisp2   __ctype        __ctype1       __ctype2
__ctype_       __curproc      __cursize      __cvt          __cwd
__cyt          __d            __date         __date_        __day
__days         __dbargs       __dbsubc       __dbsubn       __dcvtdisi
__dcvtst0      __dcvtst0a     __debug        __delay        __delcha
__dellin       __devclo       __devcnt       __devope       __devrea
__devsw        __devwri       __dir          __div0         __dmovtmpessi
__doexec       __doprnt       __dorigin      __doscan       __dosioctl
__dospawn      __dosret0      __dosretax     __dtab         __dtoxmode
```

```
__dtoxtime     __dumpdev       __ec_qui        __echoit        __edata
__end          __endope        __endopen       __endwin        __entry
__eofflag      __eptr_         __eqname        __err           __error
__errormsg     __etext         __execute       __exit          __expjmptab
__fac          __fassign       __fbuf          __fcmp          __fcsp
__fctmp        __fctopst       __ffexpm1       __ffree         __filbuf
__file         __filename      __find          __findbu        __findfi
__findio       __findiop       __first_        __fiscal        __fixdel
__fixup        __flag          __flsbuf        __fltin         __fltout
__fltused      __fmalloc       __fmode         __fmsize        __fndptr
__fnum_        __foo           __forcdecpt     __forceh        __forkptr
__forptr       __fpemulator    __fperr         __fpinit        __fpmath
__fpsignal     __fptaskdata    __fptostr       __fptrap        __freebuf
__ftbuf        __ftime         __ftoi          __ftol          __gd_cnt
__gdstat       __gdstra        __gdtab         __gdup          __gdutab
__getccl       __getrnge       __getst         __getstream     __gtstat
__gttab        __heap          __help          __hlmode        __hmstat
__hmutab       __hpstat        __hputab        __hsstat        __hstab
__htclos       __htstat        __htstra        __httab         __i
__i_size       __id_cha        __id_lin        __ifile_        __ifptr
__imok         __in_           __indefinite    __infinity      __init
__init_c       __init_k        __innum         __inode         __input
__inscha       __inslin        __insmod        __instr         __intr
__io2_         __io3_          __io_           __iob           __iob2
__iobuf        __iomode        __isindst       __kdsa          __kdsd
__kisa         __kisd          __kl_tty        __kpmode        __l
__lastbu       __lastbuf       __lastdate      __lastiob       __lbolt
__ldhead       __ldiv          __lib           __line_a        __line_f
__ll_mov       __ll_ref        __lmul          __lnjmptab      __lnkerr
__lnum_        __loc1          __logemax       __logjmptab     __logn_
__lpdays       __lptr_         __lrem          __lshl          __lshr
__lstptr       __main          __maperror      __mapit         __mausmap
__max          __mba_cnt       __mbacf         __mbadev        __mcount
__membrk       __mount         __move          __msginfo       __msgque
__msgtab       __mtab          __multi         __n             __n_n
__n_name       __n_nptr        __n_offset      __n_zeroes      __name
__name_        __namef_        __namptr        __nargs         __new_tt
__nfree        __nite          __nmalloc       __nmsize        __nogud
__nsbrk        __nswap         __null_         __nullsy        __object_
__objectq_     __offset        __ofill         __ogetf         __ogetv
__op           __openfile      __openi         __oprt          __oread
__oserr        __osfile        __osmajor       __osminor       __out_
__outch        __outcha        __output        __ova           __ovd
__ovfl         __panicstr      __parptr        __pause         __pbottom
__pbuf         __period        __pfast         __pfile         __piby2
__pickup       __pipe          __pipedev       __pos           __positive
__post         __pre           __pre_addr      __pre_cnt       __print
__proc         __profil        __prs           __psp           __ptacct
__ptr          __ptrandisp1d   __ptrandisp1s   __ptrandisp2d   __ptrandisp2s
__putcha       __putchar       __pwr_clr       __rawmod        __rawmode
__rc           __rdtrd         __readop        __reason        __req_
__res_fl       __res_flg       __reset         __rfstat        __rftab
__rkstat       __rkstra        __rktab         __rl_cnt        __rlatof
__rlfltpr      __rlprint       __rlstat        __rltab         __rootdev
```

```
__rpop          __rpstat        __rpstra        __rptab         __rpush
__rtbignan      __rtchsifneg    __rtifprojnpop  __rtifprojpop   __rtindfnpop
__rtindfpop     __rtinfnpop     __rtinfpop      __rtnospop      __rtnospopde
__rtonenpop     __rtonepop      __rttosnpop     __rttosnpopde   __rttospop
__rttospopde    __rtzeronpop    __rtzeropop     __sbmap         __sbrk
__sbrkslop      __sbrpte        __sbuf          __sccsid        __scrdow
__scroll        __sctab         __sdget         __sema          __seminfo
__setbuf        __setdta        __sethl         __setmod        __setwin
__shmem         __shminfo       __shove         __showst        __sibuf
__sinhjmptab    __sinjmptab     __size          __smbuf         __smovtmpessi
__sobuf         __sorigin       __sp_           __spar          __sprint
__sprintw       __sqrtjmptab    __sscans        __st            __start
__statefile     __stbuf         __stdbuf        __stime         __stkgro
__stmax         __stop          __string        __stringq_      __strinit
__strout        __subop         __sum           __super0        __super1
__swapdev       __swapmap       __swbuf         __swbuf1        __swbuf2
__swplo         __syncmo        __syscal        __syserr        __sysinfo
__sysname       __tanhjmptab    __tanjmptab     __tbls          __text
__time          __tmclos        __tmp           __tmstat        __tmstra
__tmtab         __token         __tolower       __torigin       __toupper
__trandispl     __trandisp2     __trap          __tscrol        __tsstat
__ttread        __ttstat        __ttwrit        __tty           __tty_ch
__tvorig        __tzflag        __u             __uba_cnt       __ubacf
__uldiv         __ulmul         __ulrem         __ulshr         __umaskval
__umemvad       __unctrl        __user          __userid        __utsname
__uun_          __v             __var           __verify        __wait
__whptr         __win           __window        __writec        __wrtchk
__wtmp          __x             __x25info       __xcend         __xcleanup
__xcstart       __xed_          __xflsbu        __xfunc         __xiend
__xistart       __xpop          __xpush         __xxxx          __ytoxjmptab
__yystype_
```

5. HISTORY

For the past few months, a group of people from Operating Systems and Systems Languages have been discussing the naming conflict problem and the possibility of adopting a naming convention to solve it.

The simplest approach seemed to be a convention in which names are unique to each product - that is, they are somehow qualified by the product name. If the qualification could be guaranteed not to conflict with any existing name, the convention could be implemented very easily.

Furthermore, if special names began with a character that users were unlikely to choose for their own names, we could minimize the possibility that a user's symbol would accidentally collide with a hidden name.

The first naming convention proposal suggested using a dollar sign at the beginning of each hidden name, and modifying the compilers to allow dollar signs to begin names. Unfortunately the dollar sign causes grave problems for a number of utility programs, including some assemblers and debuggers. Even more unfortunately, almost all the other special characters caused worse difficulties.

Consequently, we resorted to the underscore, which was already in use for a similar purpose. Because C already uses a single underscore for all public names, we decided to use two. The qualifier for a name would consist of two underscores followed by a single character identifying the associated product.

Since there was the possibility that names with this form might already exist, we gathered namelists from existing products and libraries to form the "exceptions" list. No name appearing on the exceptions list may be used for a new purpose, because a conflict might arise.

After a review by group leaders and other members of the development staff, we put the naming rules and exceptions list into this document for distribution.

6. PITFALLS

It is possible that some existing names in different products may already conflict. Where possible, we should revise such names to follow this convention. Full upward compatibility may preclude this possibility. If so, resolution of such problems is beyond the scope of this document.

Although there is a category for names which are "common", it is not at all clear what criteria should be used for classifying such names. Furthermore, there are technical issues to be resolved, such as calling protocols for "common" library routines. If anyone is interested, this might be a good topic to pursue.

Another problem is the type and number of significant characters in a name. Microsoft's systems products allow very long names with many significant characters, but we can't guarantee that they'll always be used. (For example, we have OEMs that have ported our compilers to other environments for cross-development.) Underscore characters or long names may cause problems for foreign linkers. If you're particularly concerned about portability, you should restrict your external names to six characters, including the three-character prefix. If you're cautious but not fanatical, you should restrict your external names to the eight characters supported by most versions of UNIX. A macro preprocessor can be used to remove the underscores if necessary.

7. ACKNOWLEDGMENTS

Vic Heller, Andy Padawer, and Barry Shaw participated in the discussions that led to this standard. Dave Weil and Andy Padawer prepared the exception lists. Ross Garmoe suggested several improvements to this presentation. Thanks to all!